

コンピュータと機械システム

下条誠*

*電気通信大学

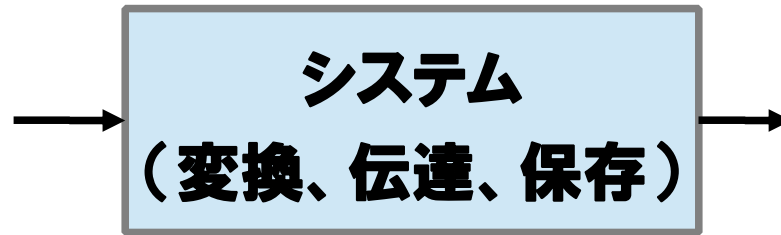
達成目標

コンピュータによる制御について理解し、状況に応じた動作を行う電子機械の構成法が分かる。

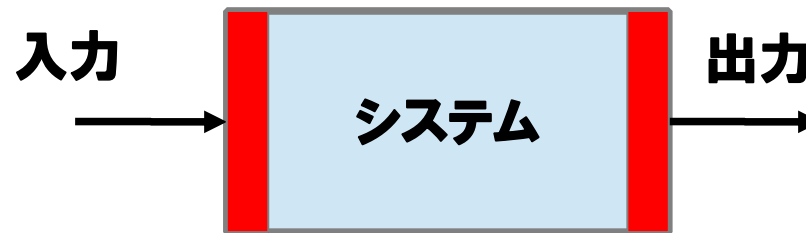
システムとインタフェース

相互に影響を及ぼしあう要素から構成される、まとまりや仕組みの全体

入力
物質
エネルギー
情報

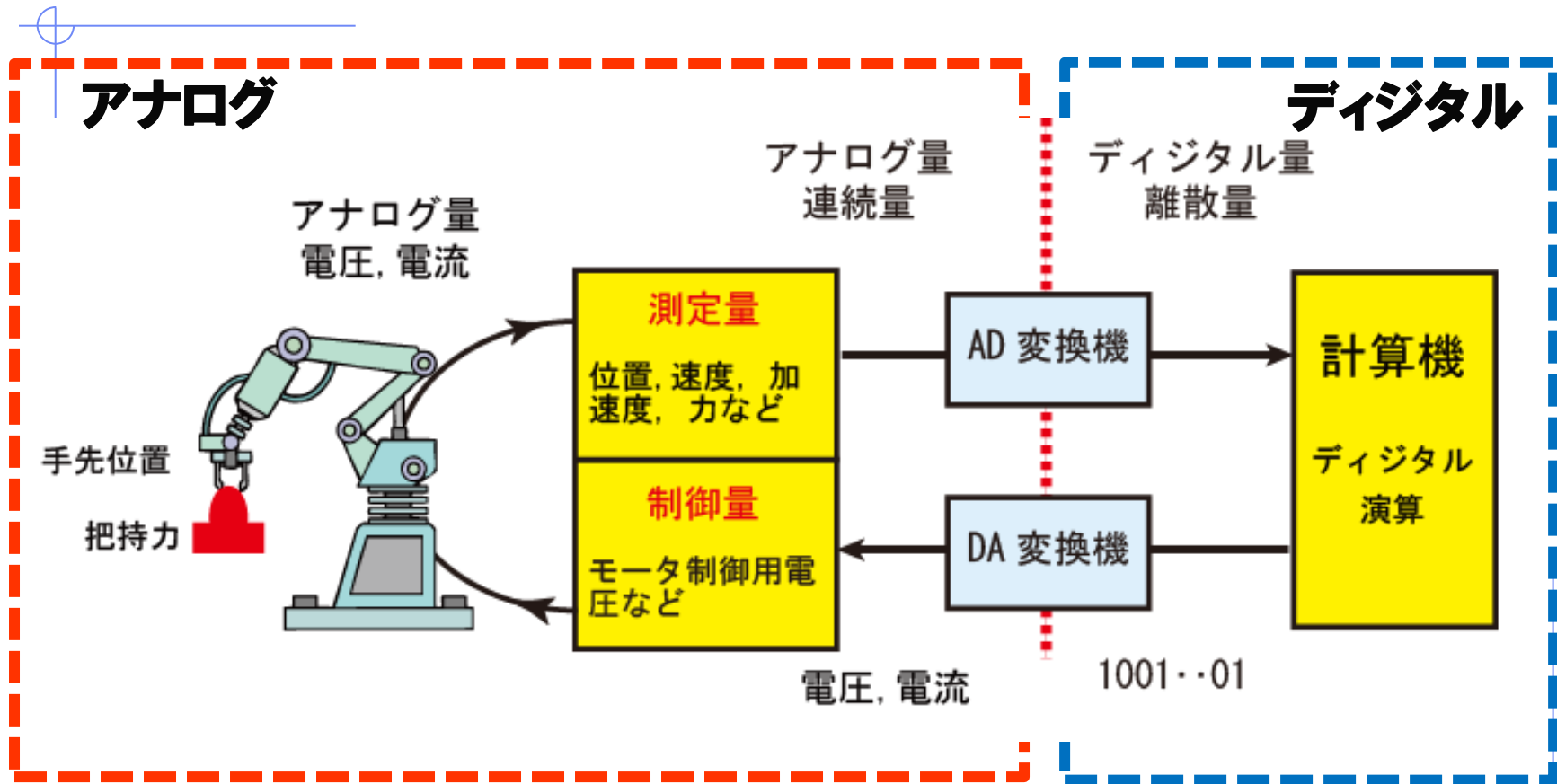


出力
物質
エネルギー
情報



インタフェース: 物質、エネルギー、情報の授受を行う、その接触面

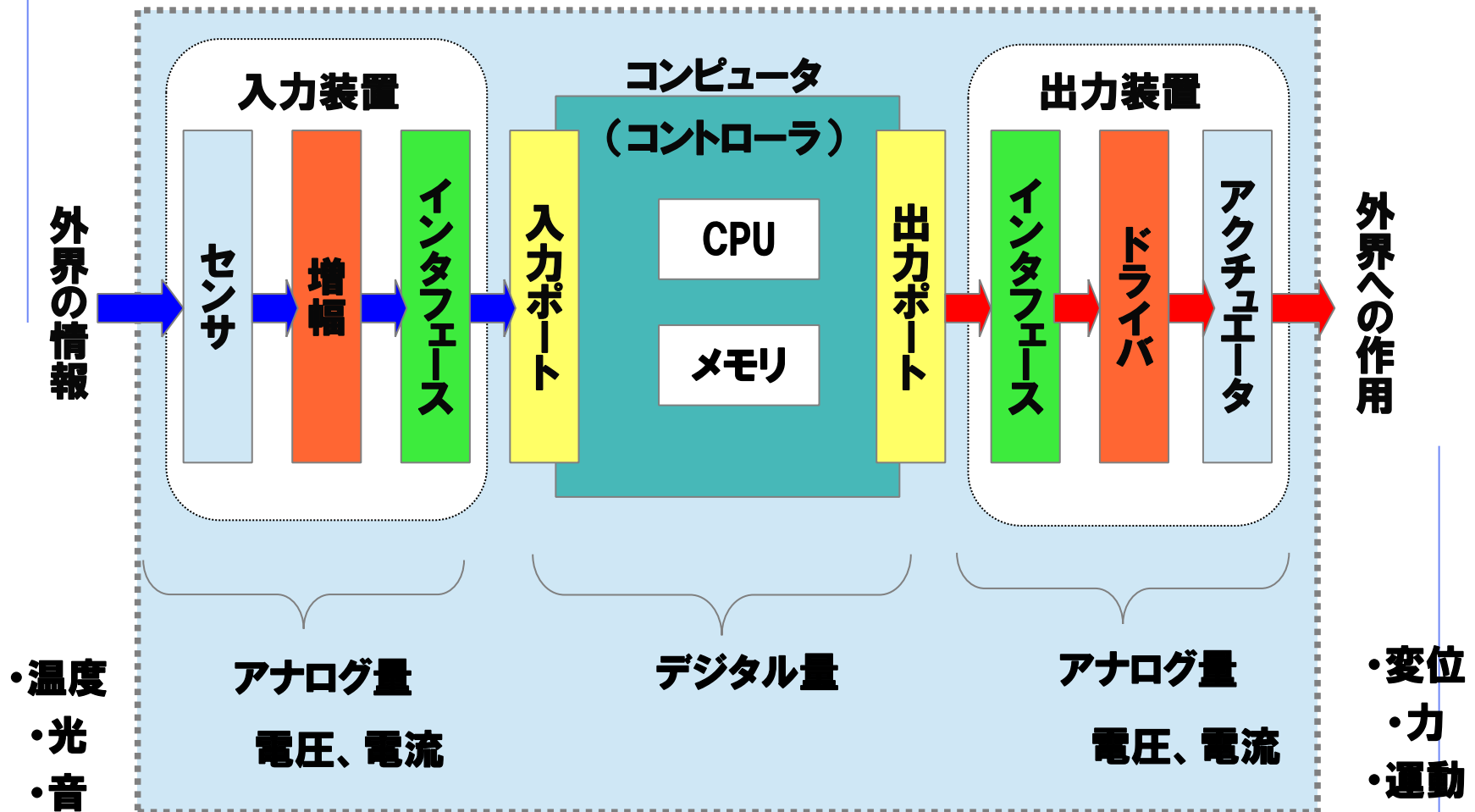
メカトロニクスシステム: アナログとデジタル



現実の世界は,
アナログ情報

計算機の世界は
デジタル情報

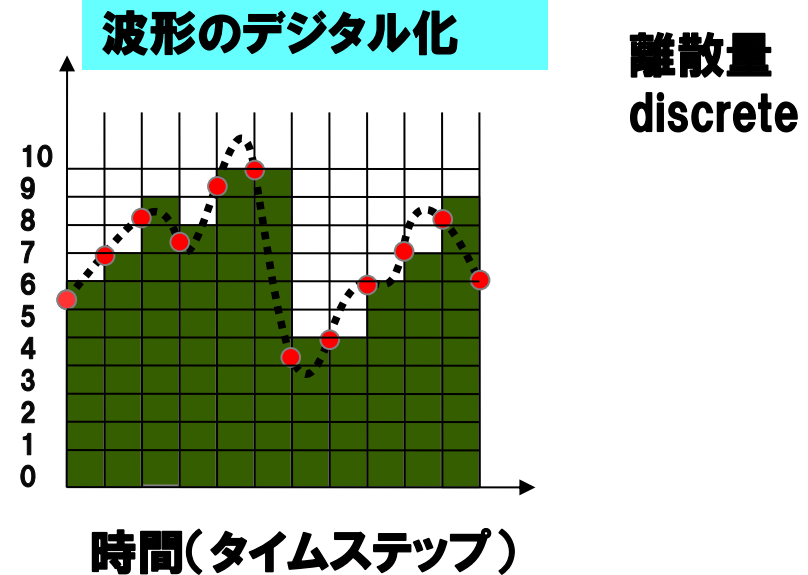
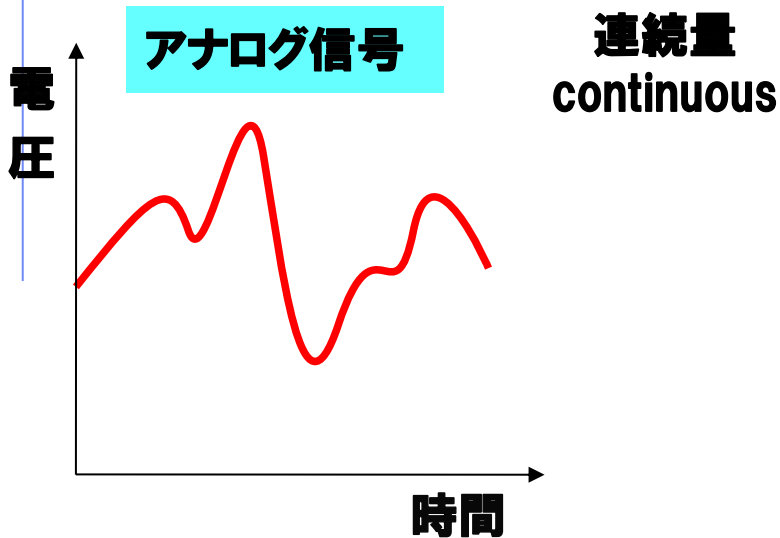
メカトロニクスシステムの基本構成例



アナログとデジタル

アナログ:連続量

デジタル:量子化、離散化した量

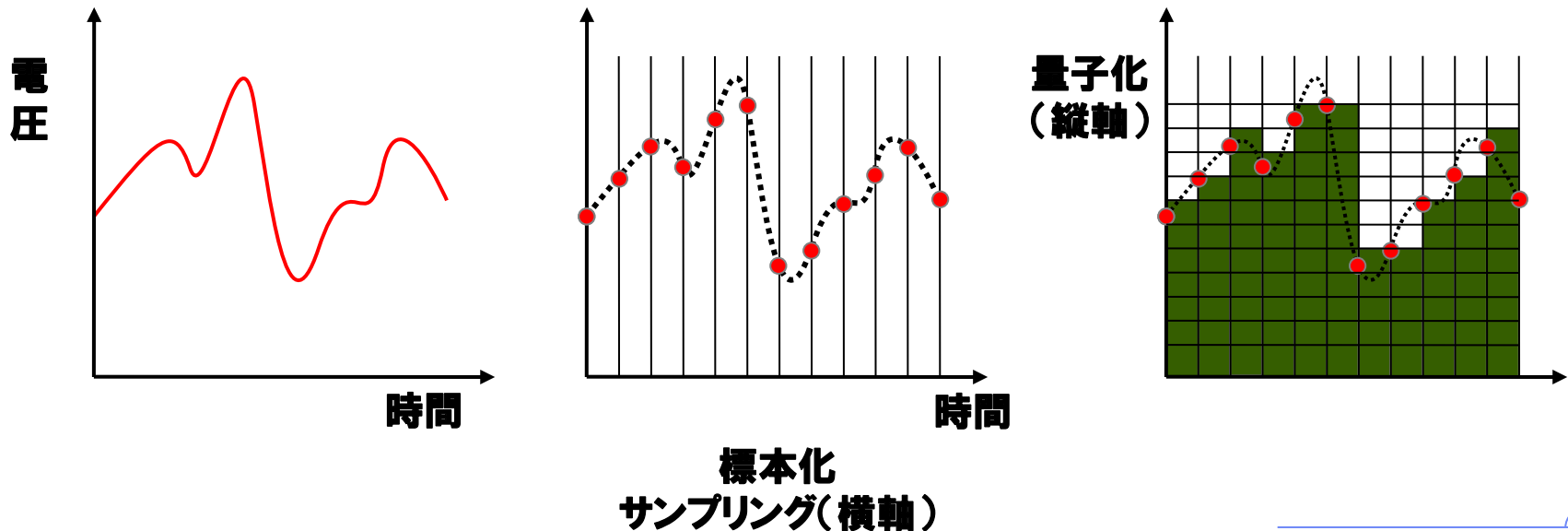


1. 自然界のものは基本的にアナログ。
2. それをデジタル化してコンピュータで扱う。
3. 自然界に戻す際に、またアナログへ

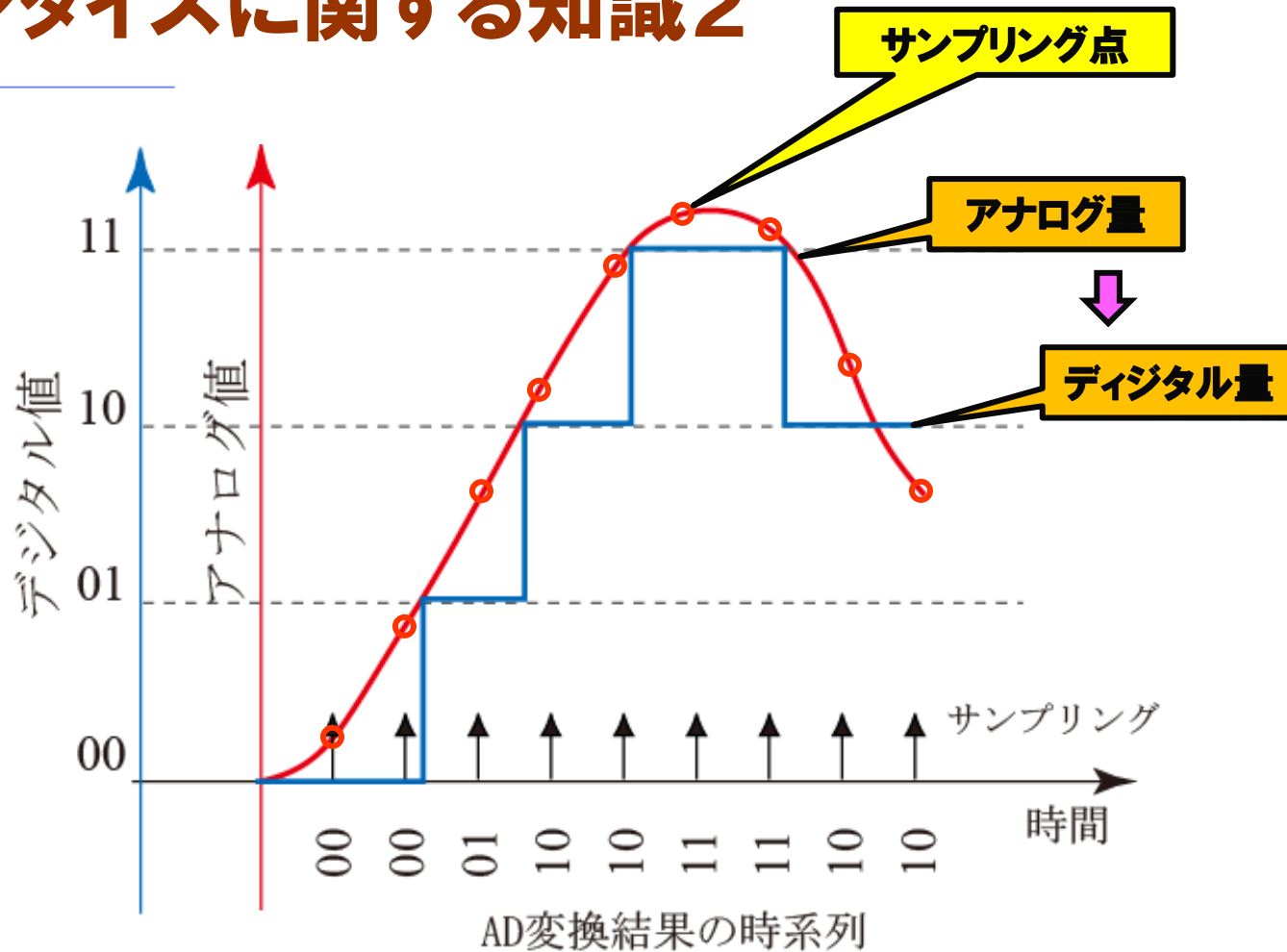
デジタイズに関する知識 1

AD変換

- ✓ 標本化 sampling: 一定時間ごと(周波数)に離散化
- ✓ 量子化 quantization: ある単位(量子)で区切る
- ✓ 分解能: 縦軸をどれだけ細かくとるか(ex.10bit→1024段階)
- ✓ 量子化誤差: 元の信号と量子化後の信号との差



デジタルズに関する知識2

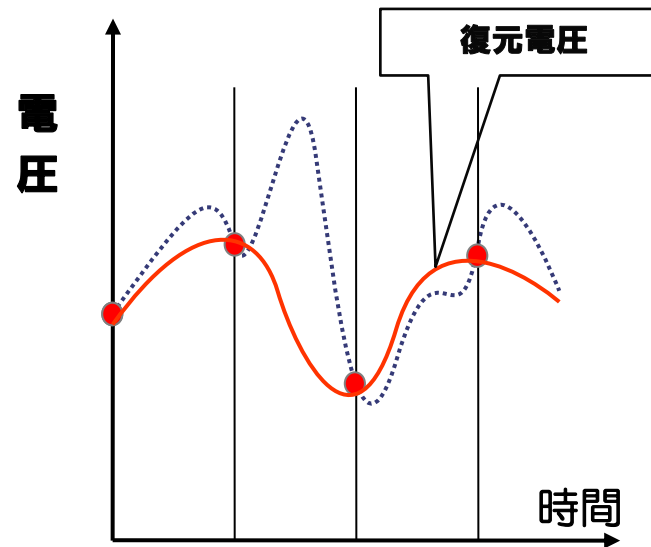
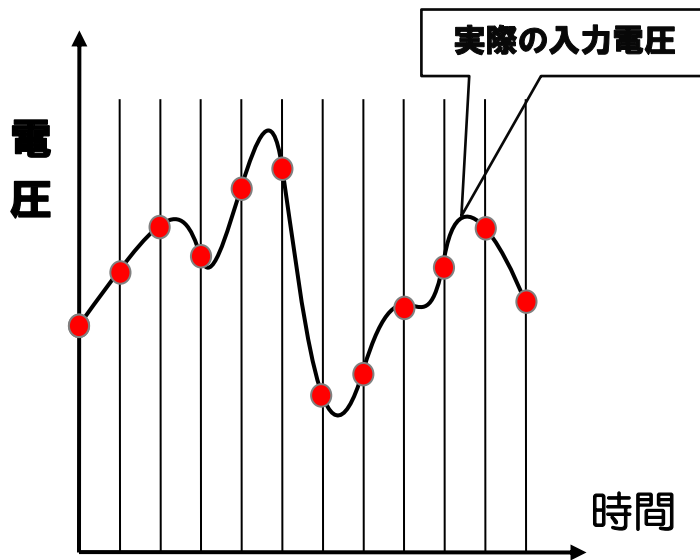


アナログ量→デジタル量(2ビット)へ変換例

デジタルに関する知識 3

DA変換 デジタル→アナログ電圧

サンプリング周波数が低いと元の信号の復元困難



サンプリング周波数の半分の周波数まで再現可能

ex) CDのサンプリング周波数44kHz
人間の可聴領域 20Hz~22kHz

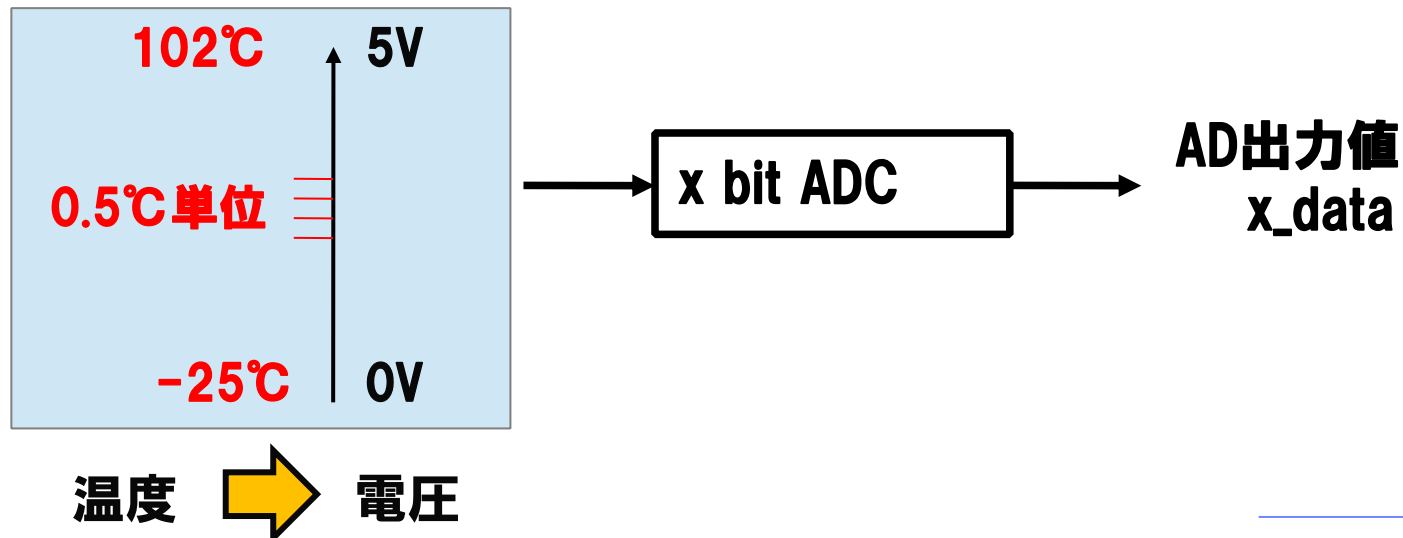
AD変換器にまつわる計算

温度センサ(出力範囲:0~5V,-25°C~102°C)の値を0.5°Cごとに測りたい。

問:

1. 電圧は何mV単位で測る必要があるか?
2. AD変換器の分解能は最低何bit必要か?
3. AD出力値 x_data から温度 [°C] への変換式を示せ

温度センサ



AD変換器にまつわる計算2

温度センサ(出力範囲:0~5V,-25℃~102℃)の値を0.5℃ごとに測りたい。

問:

1. 電圧は何mV単位で測る必要があるか?
2. AD変換器の分解能は最低何bit必要か?
3. AD出力値 x_data から温度 [℃] への変換式を示せ

答え

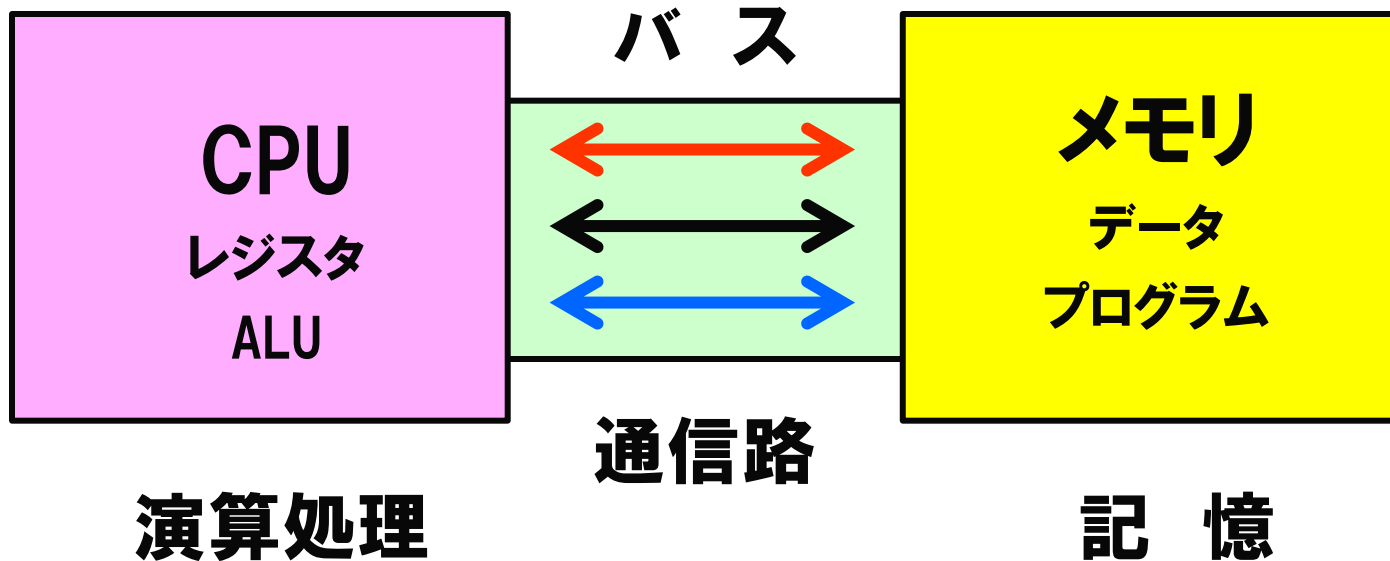
問1. $5[V] \div \frac{102 - (-25) + 1}{0.5} = 19.5[mV]$

問2. 温度:-25℃~102℃で128段階の変化。これを0.5度刻み

$$128 \div 0.5 = 256 = 2^8 \Rightarrow 8bit$$

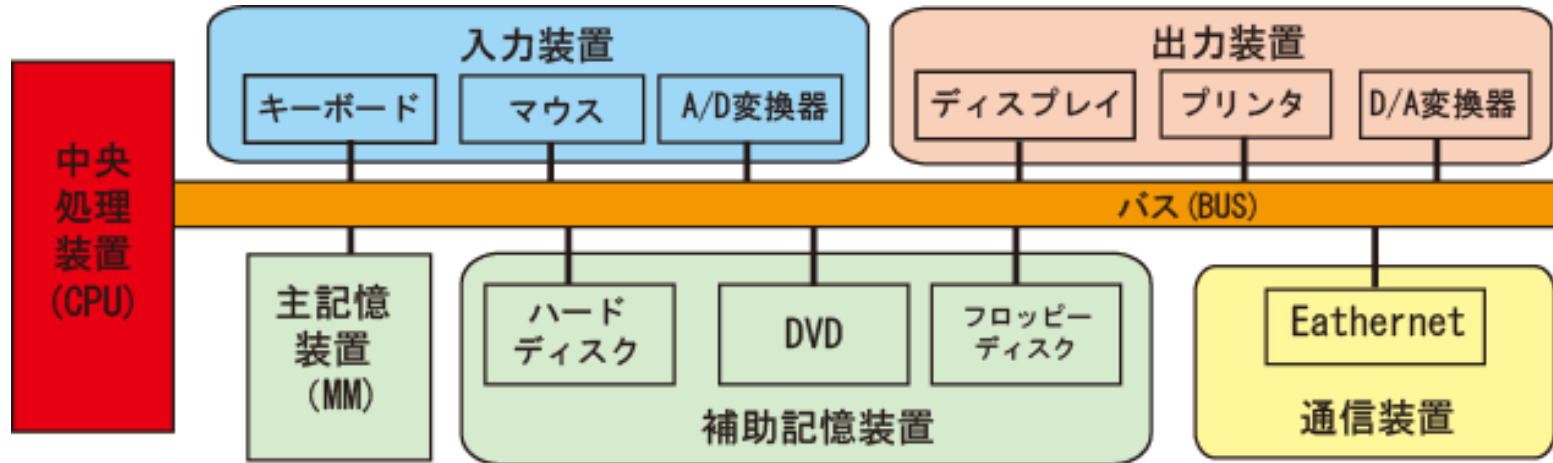
問3. $x_data \times \frac{128}{256} - 25 = 0.5 \times x_data - 25$

コンピュータの基本ハード構成



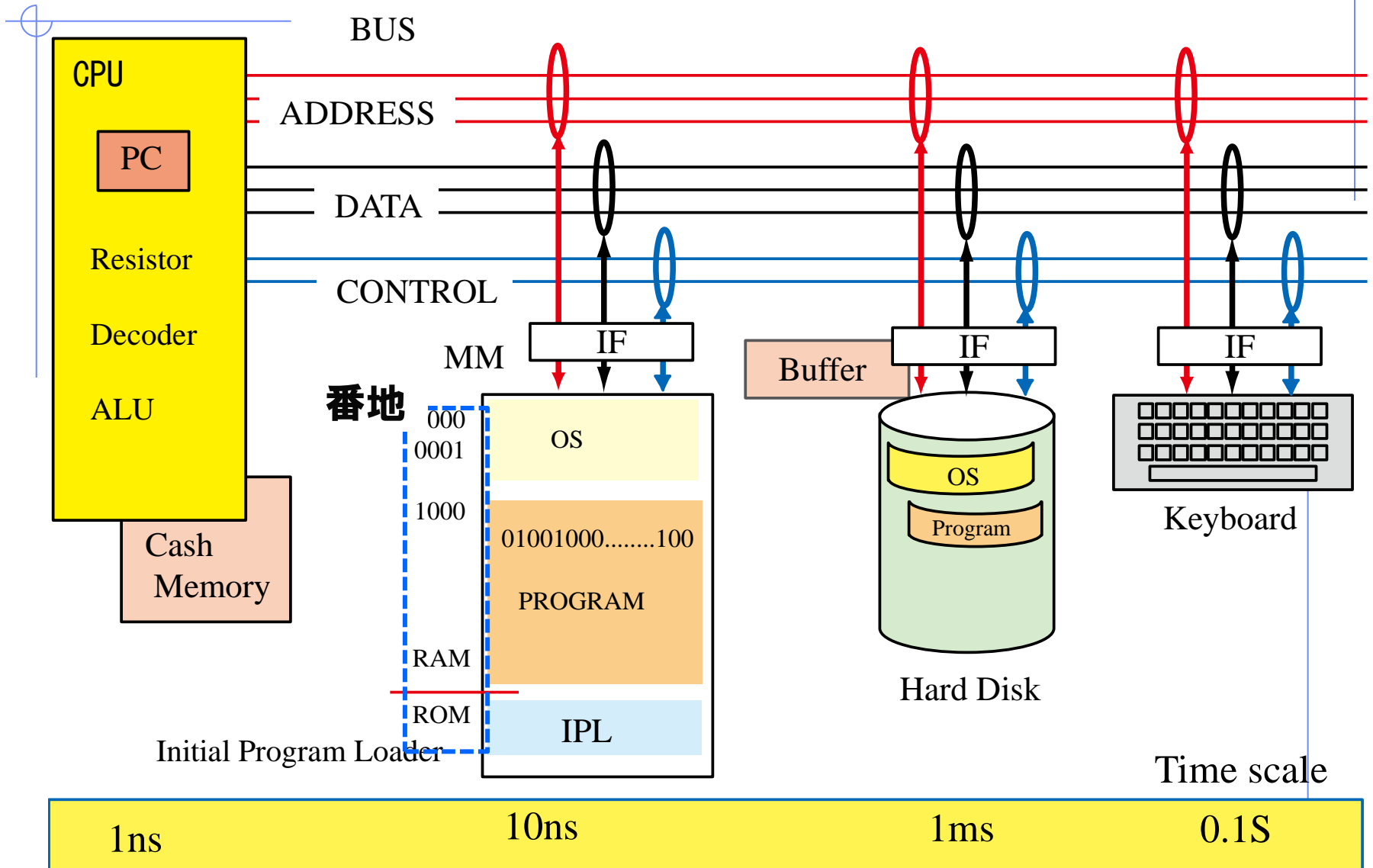
1. CPU は, コンピュータの管理と演算処理を行う
2. メモリは, データを記憶する(プログラムもデータの種類)
3. バスは, CPU とメモリの間をつなぐ通信路(配線の束)である

コンピュータの基本ハード構成



1. メモリの中の全記憶場所には**アドレス(番地)**がついている。
 2. CPU は、データの読み出し／書き出しは、**アドレス**を用いる。
 3. CPU は、プログラムの通り実行し続ける。しかし、**突発・緊急事態**に対して**処理**を行うようになっている。これが**割り込み処理**である
- **ハードディスク、キーボードなどの入出力装置もデータの出し入れを行うメモリの延長と考えられる。よってそれぞれにアドレスを割り付けバスに接続する**

CPUと外部装置との結合(バス構造)



バスの役割

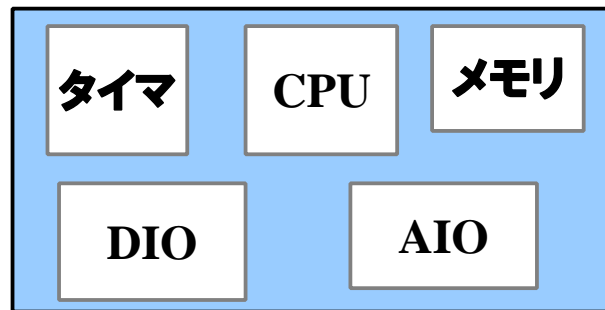
- **アドレスバス**:データのアドレス(番地)を伝える。
データの入出力を行う場合, まずCPU から必要とするデータの格納場所のアドレスをアドレスバスに送る。
- **データバス**:CPU と他装置(メモリ, I/O)への「読み／書き」するデータを送る。
- **制御バス**:制御信号を送る。
例えば, CPU からの入出力 (Read/Write) 指令, また割り込み信号等が伝えられる。

動作の説明

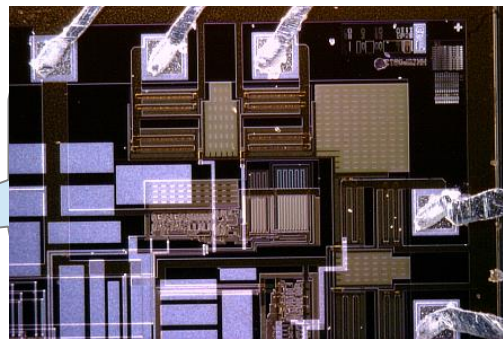
1. CPU はデータの**アドレス**をアドレスバスに送り出す
2. CPU はデータの**Read/Write 指令**を制御バスに送り出す
3. アドレスに該当する装置は制御バスの指令信号による動作を行う
 - (a)**Read 指令**:アドレスにあるデータをデータバスに送る
 - (b)**Write 指令**:データバスにあるデータをアドレスにしまう
4. CPU はデータRead 指令の場合データバスに送られたデータを取り込む

マイコンとは

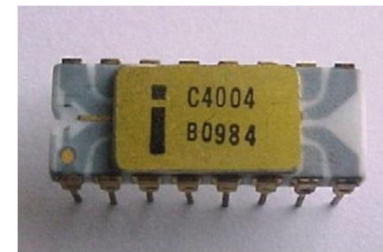
Micro control unit (MCU), Micro controller
一つの半導体チップにコンピュータシステムを集積したLSI製品
CPU、メモリ、入出力(I/O)回路、タイマ回路などを格納



H8



マイコン内部写真



世界初マイコン 4004
(1971)

マイコンの特長

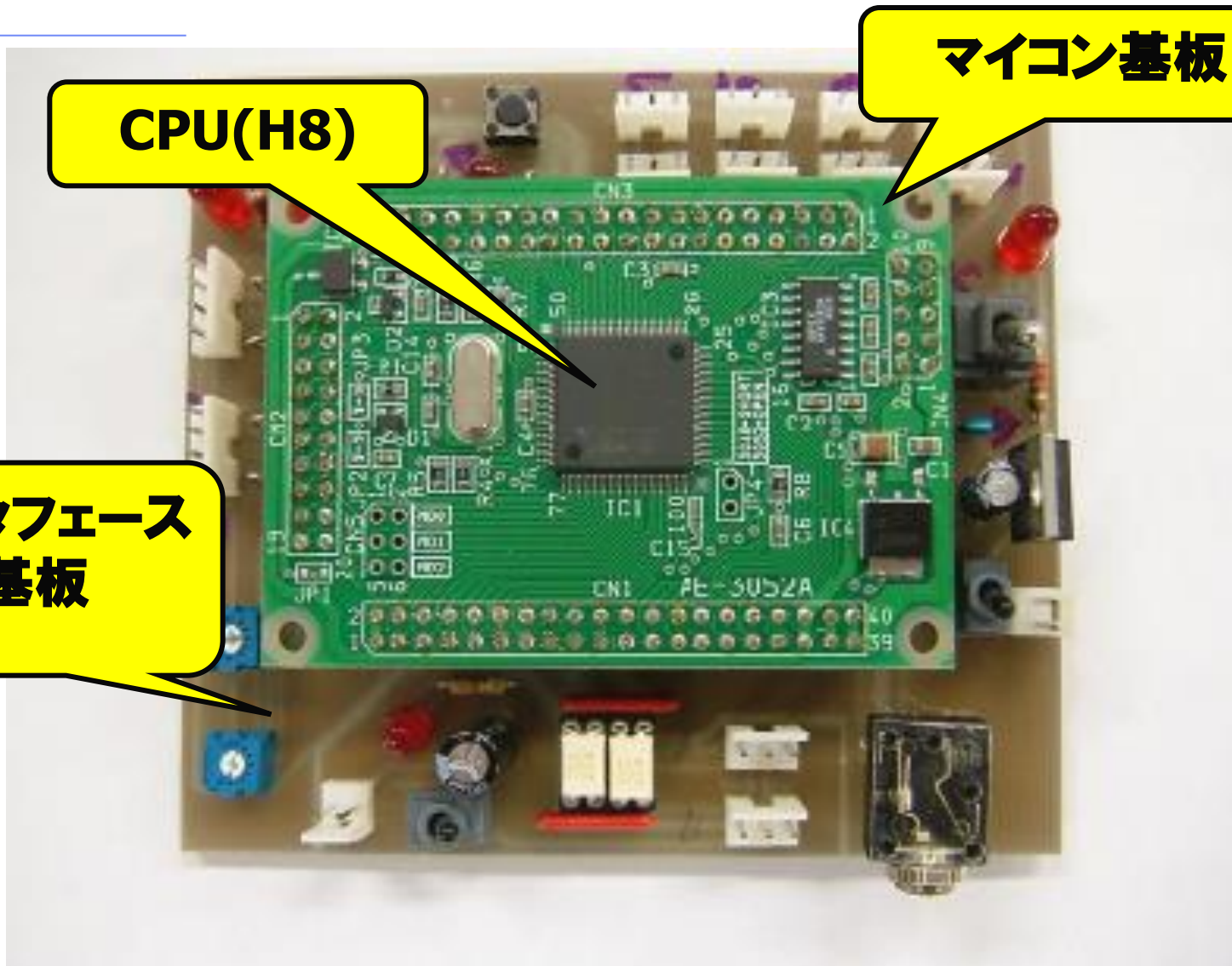
- 小型、低価格
- システム全体の基板面積や部品点数、消費電力も少ない
- 制御が複雑になるに従い、すべてを回路のみで構成する
ワイヤードロジック(ハードウェアロジック)よりも低コスト
- プログラムを書き換えることで機能変更可能

家電からAV機器、携帯電話、産業機械、自動車などほとんどの電子・機械製品に何らかの形でマイコンが組み込まれている。

種類:

Z80、H8、SH2、ARM...etc

H8マイコン1

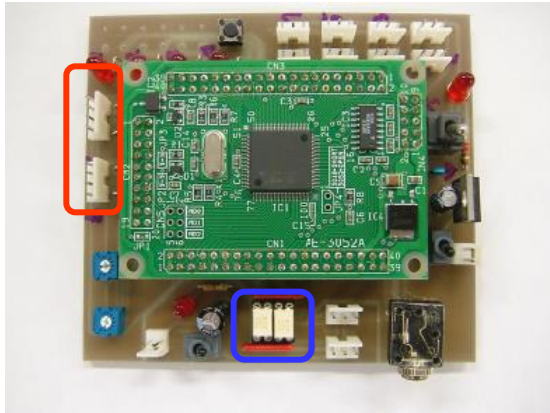


CPU(H8)

マイコン基板

**インターフェース
基板**

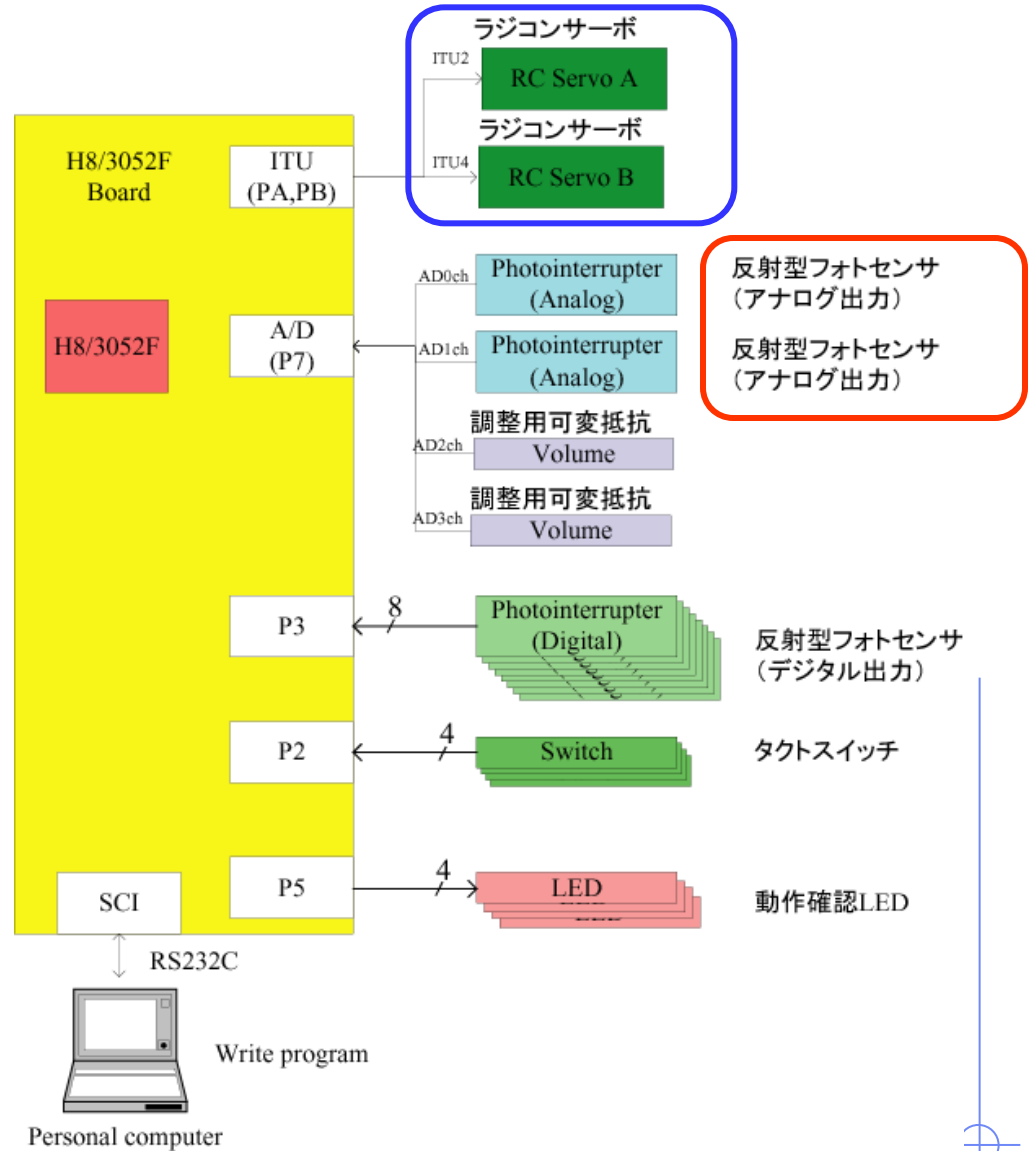
H8マイコン2



センサから情報入力



モータ制御



16進表現

表 D.4: 16 進表現

数	2 進表現	16 進表現	数	2 進表現	16 進表現
0	0000	0	8	0100	8
1	0001	1	9	0101	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

例

1. "10100011" → "1010 0011" → "A3" となる
2. "1011011111110011" → "1011 0111 1111 0011" → "B7F3" となる
3. これを 16 進表現であることを、明記するため、数値の先頭に "0x" をつける。
4. "10100011" → "0xA3" となる

2進数の表記法

1. "10100011" → "1010 0011" → "A3"となる
2. "1011011111110011" → "1011 0111 1111 0011" → "B7F3"となる
3. これを16進表現であることを、明記するため、数値の先頭に"0x"をつける。
4. "10100011" → "0xA3"となる
5. "1011011111110011" → "0xB7F3"となる

<参考> 16進数の表現方法

16進数の表現方法は、マイコンの種類、コンパイラによって異なる。
例えば、以下の表記は同じものである。

ちなみに"H"はHexadecimalの事を示す。

- H8系:H' B7F3
- 68系:B7F3\$
- 80系:B7F3H

H8のバス構造

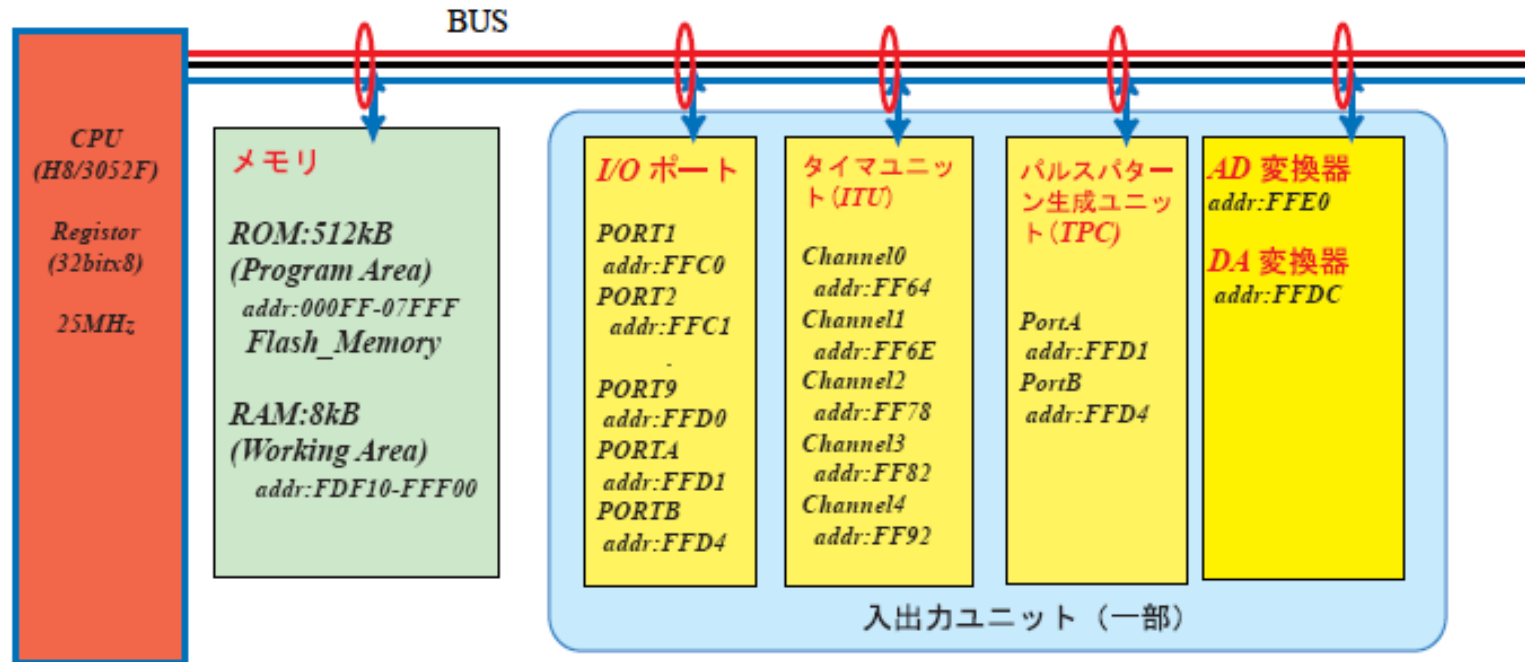
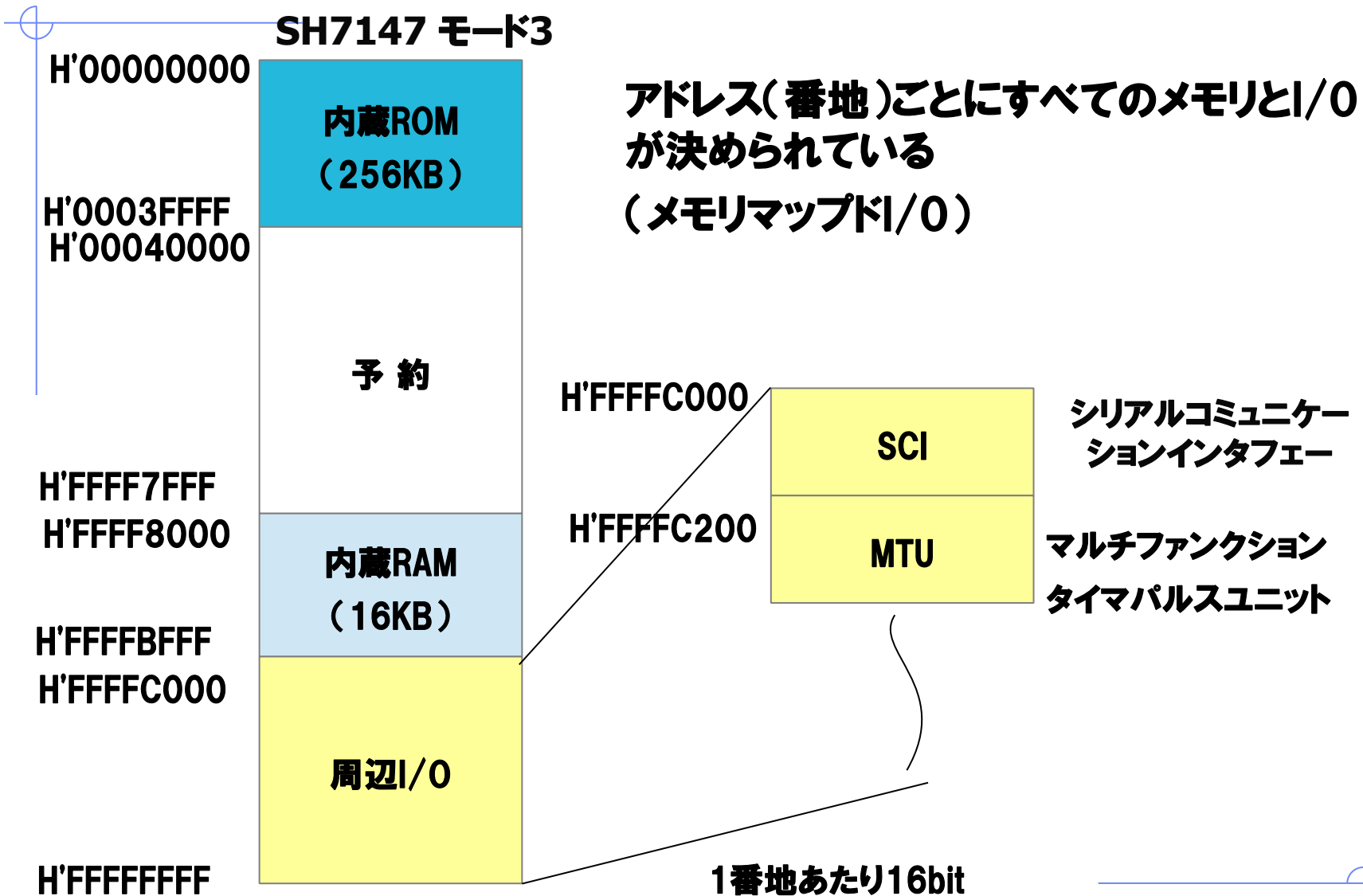


図 4.1: H8 の構成概念図

- 汎用レジスタ：16 ビット×16 本 (8 ビット, 16 ビット, 32 ビットとしても使用可能)
- 最大動作周波数：25MHz
- アドレス空間 16M バイト

アドレスマップ



信号の入出力

MSB (most significant bit) LSB (least significant bit)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

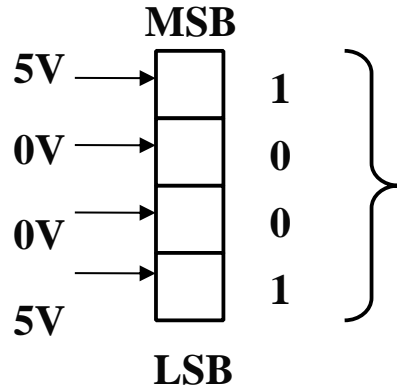
bit: 二進数の一行

1Byte = 8bit

$00000100_{(2)} = 2^2 = 4$ 16進では 0x04

$00010100_{(2)} = 2^4 + 2^2 = 20$ 16進では 0x14

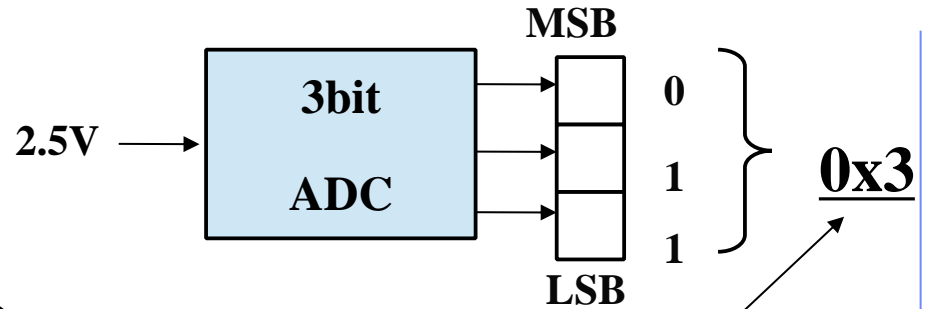
デジタル入力



出力はこれらの逆

0x9

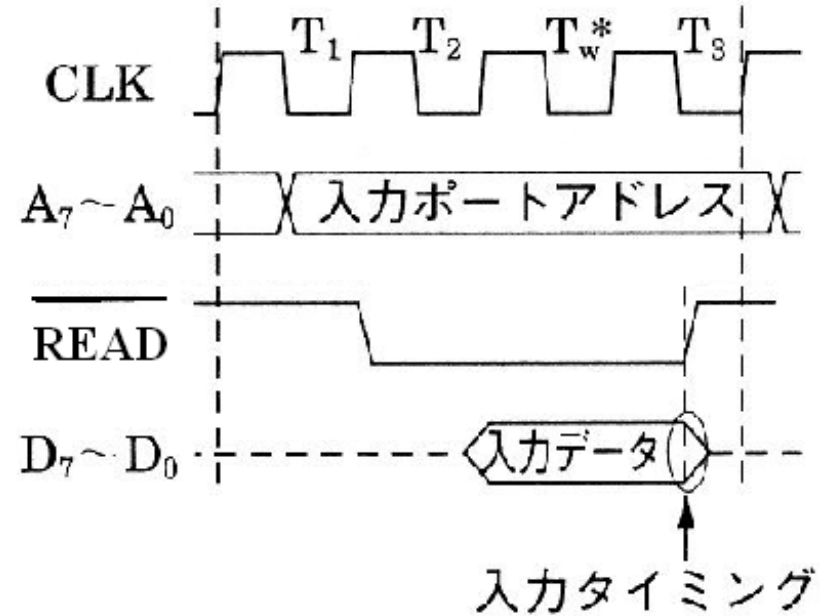
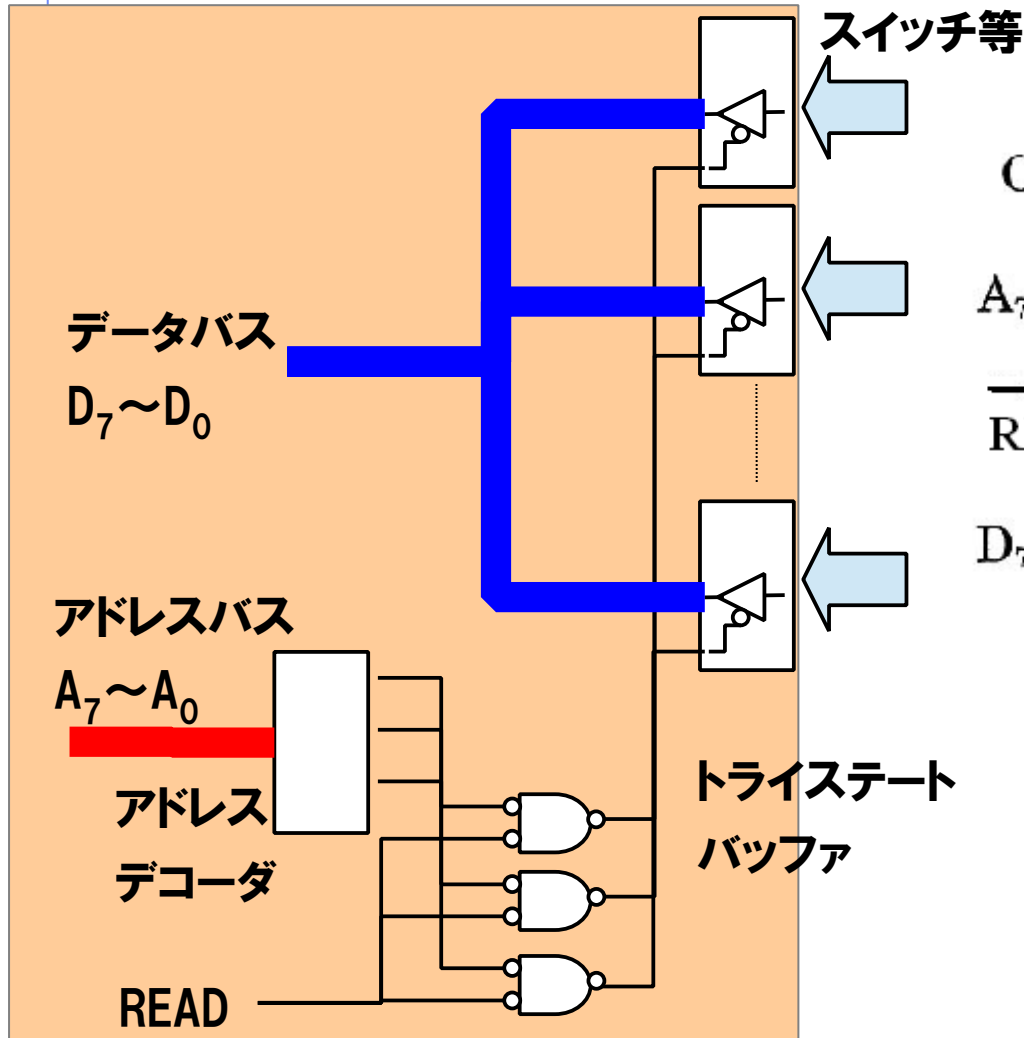
アナログ入力



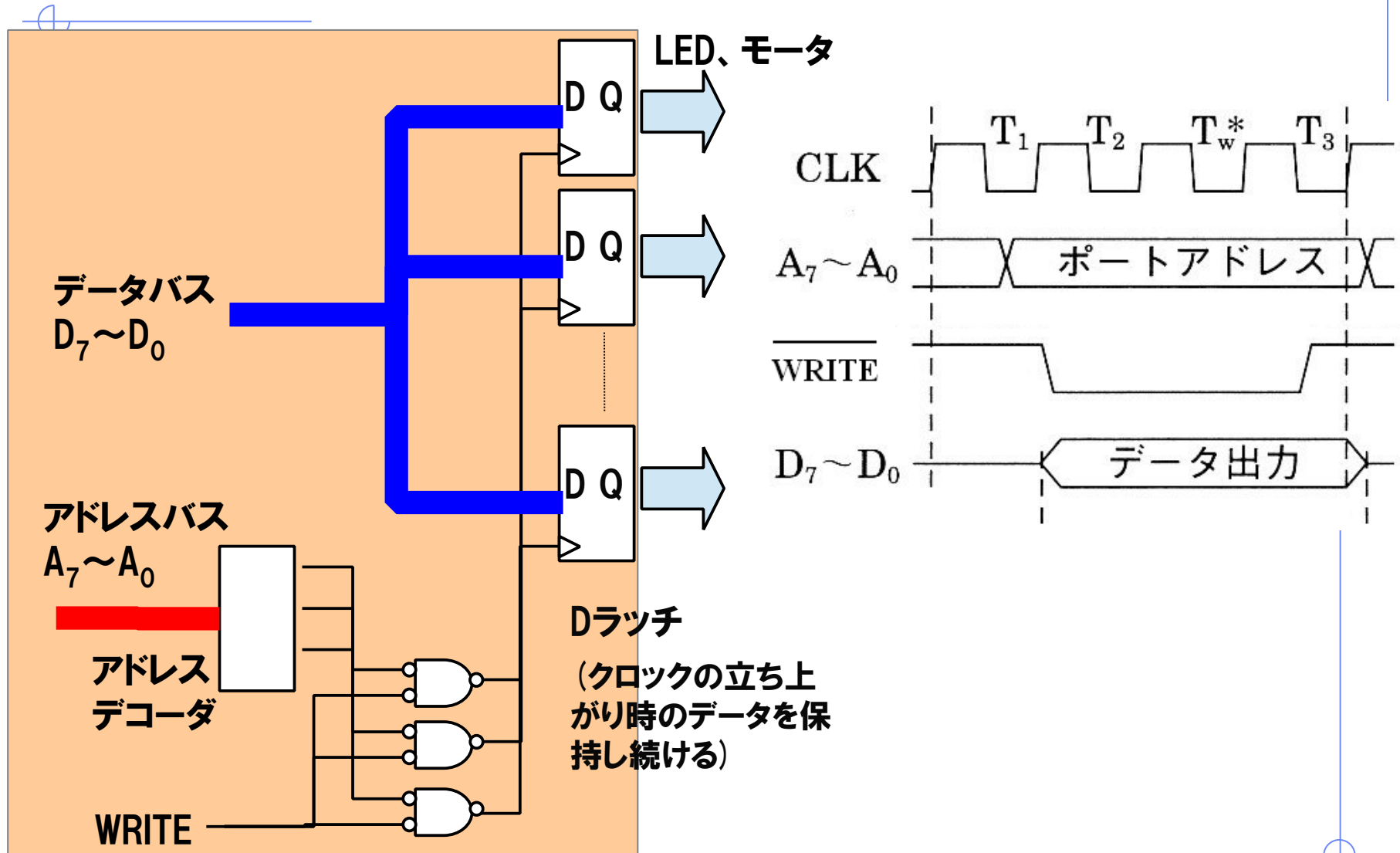
0x3

CPUが受け取るデータ

入力ポートとタイミング

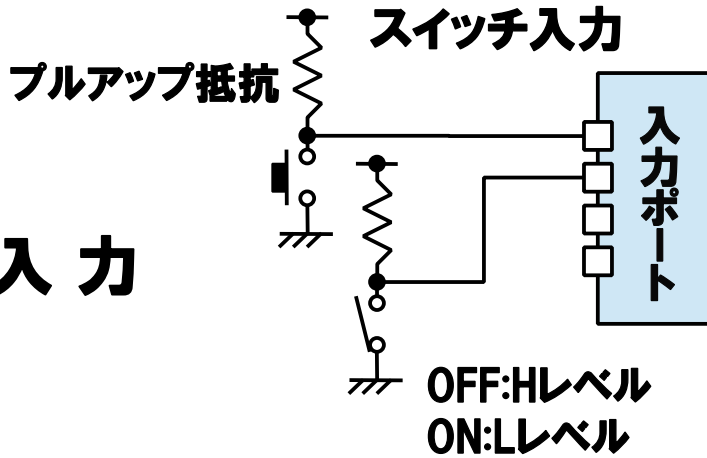


出力ポートとタイミング

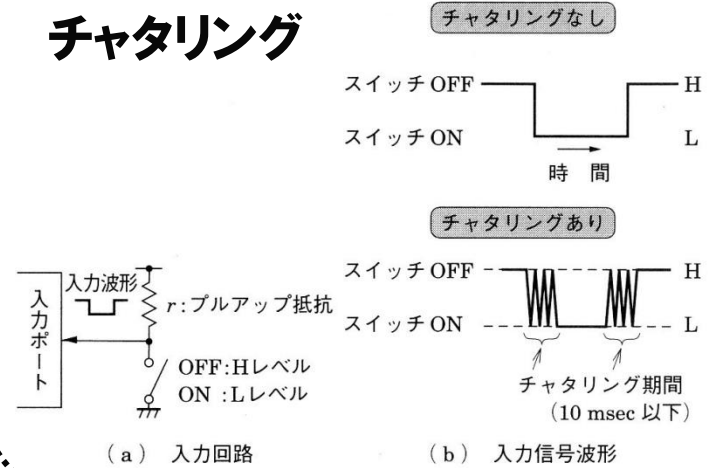


入出力インタフェース例

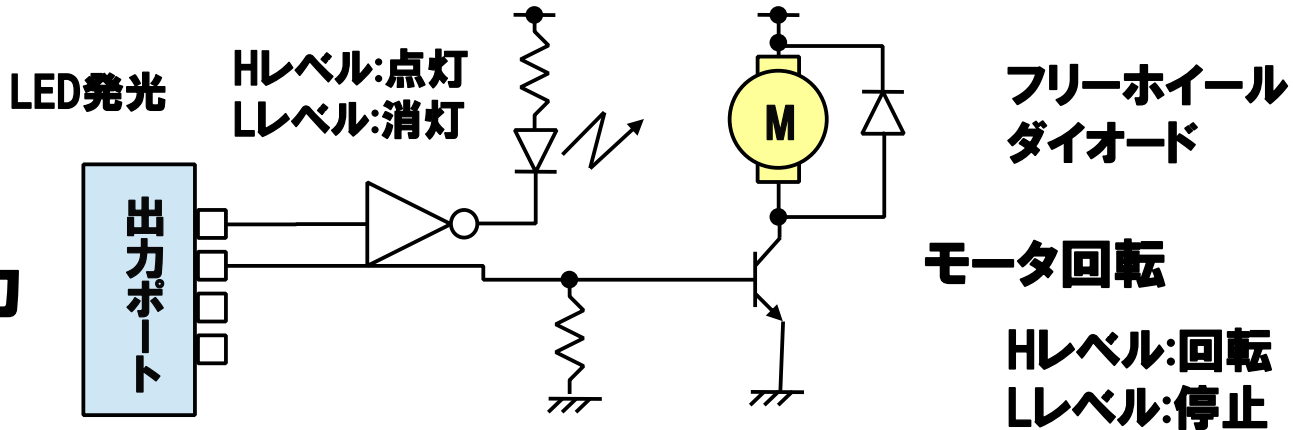
入力



チャタリング

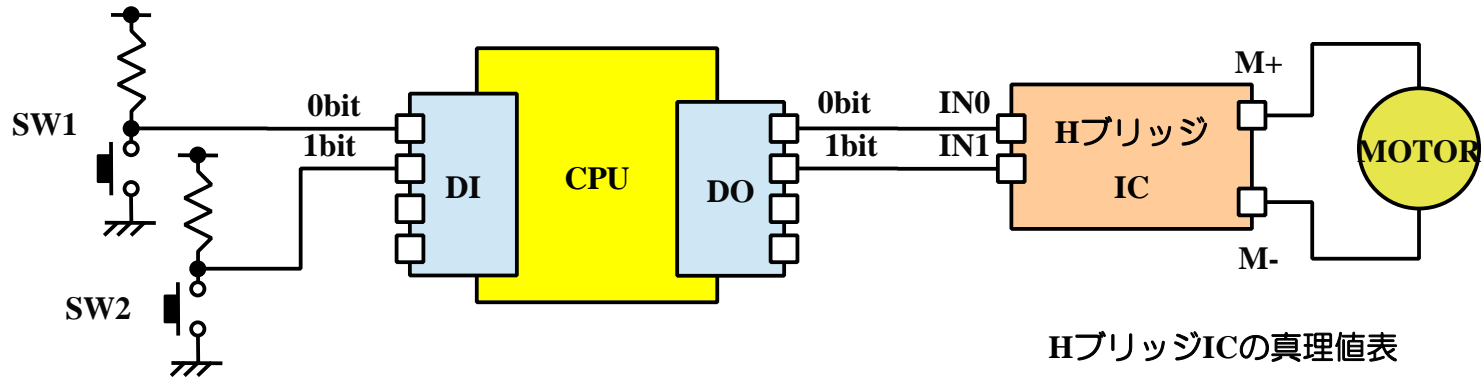


出力



例 マイコンによるDCモータの駆動

動作: SW1が, 1)ONの間モータ正転、2)SW2がONの間、モータ逆転、3)両方ONもしくは両方OFFの時はモータ停止



HブリッジICの真理値表

IN0	IN1	モータ
0	0	停止
0	1	正転
1	0	逆転
1	1	ブレーキ

```

while(1){
sw = *ADD_DI&0x03;
switch(sw)
{
case 0x00: //両方on
*ADD_DO=0x00;
break;
case 0x01: //SW2 on
*ADD_DO=0x01;
break;
case 0x02: //SW1 on
*ADD_DO=0x02;
break;
case 0x03: //両方off
*ADD_DO=0x00;
break;
}
}

```

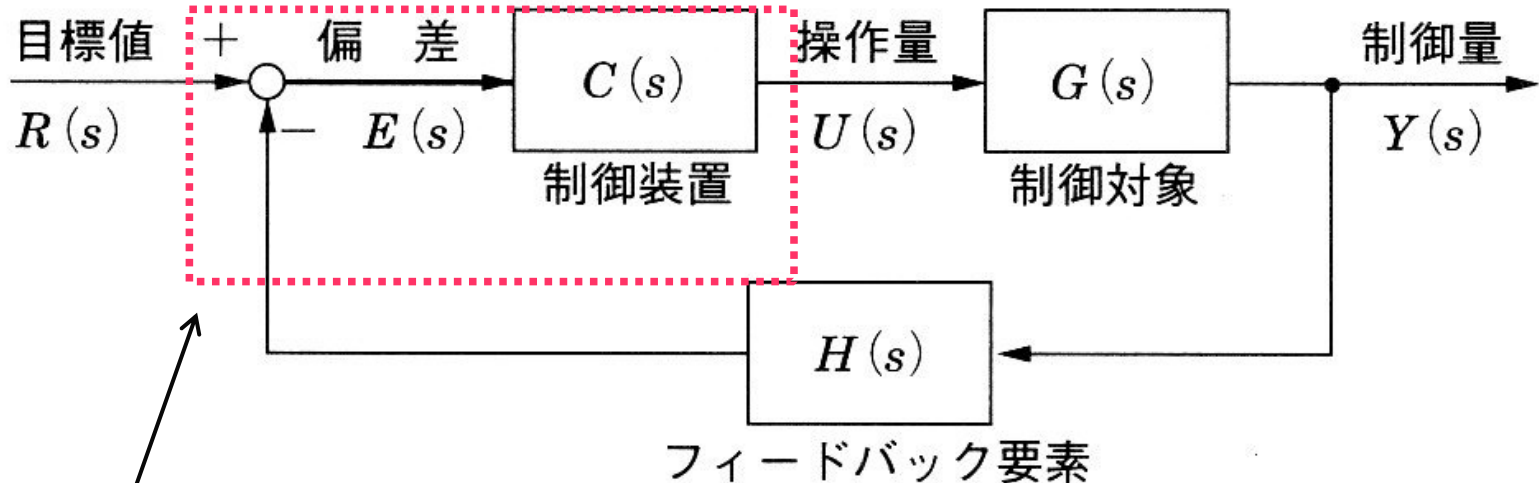
```

char *ADD_DI = (char *)0xFFFC8;
char *ADD_DO = (char *)0xFFFD8;

```

もう少し高度な制御

DCサーボモータの位置決めを考える フィードバック制御のブロック図

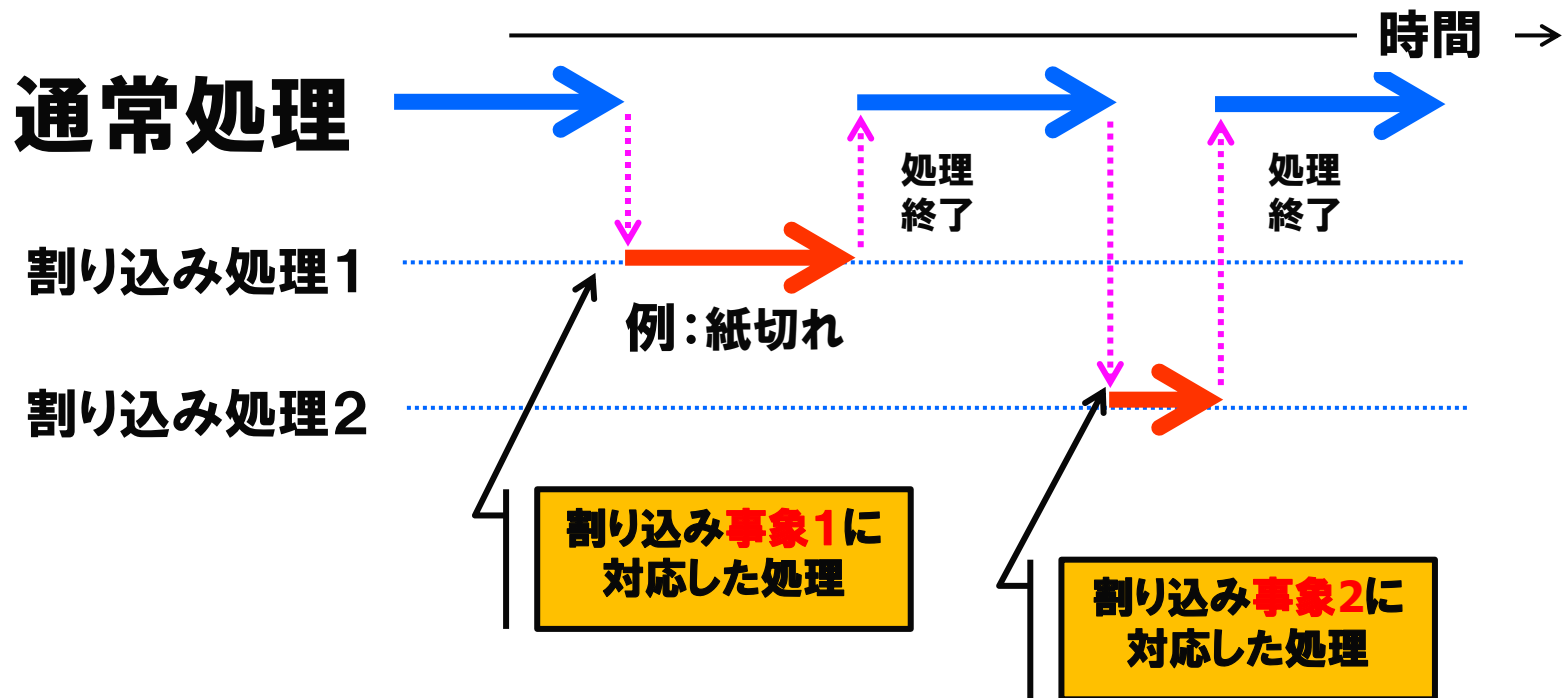


赤色の破線部がマイコンの仕事

一定の時間ごとに状態の監視、制御則の適用を行う必要がある

割り込み処理 1

突発的事態などで外部機器などからCPUに通知を行う仕組みを作り、その通知に基づき処理を行う方法である。



割り込み処理によるCPUのプログラム実行の時間的推移例

割り込み処理2

割り込みが発生したら、割り込みルーチンに飛ぶ

■外部割り込み

あるポートに反応があったらすぐに割り込み先に飛ぶ

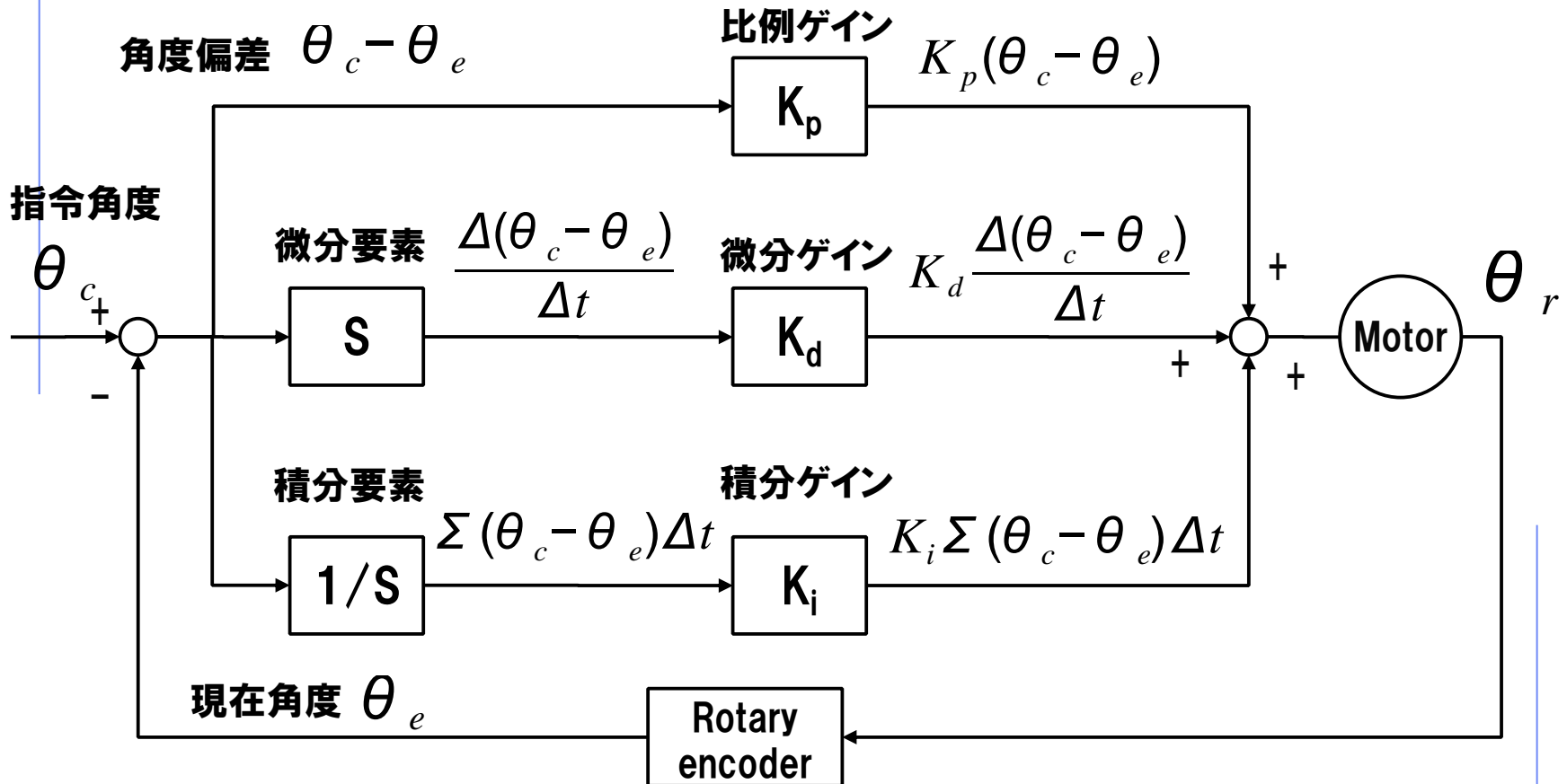
■タイマ割り込み

一定時間ごとに割り込み先に飛ぶ

サーボ制御では、割り込み先で状態の監視・制御則適用

その他、さまざまな割り込み要因があるが、優先順位が決められている

サーボ機構 位置に関するPID制御例

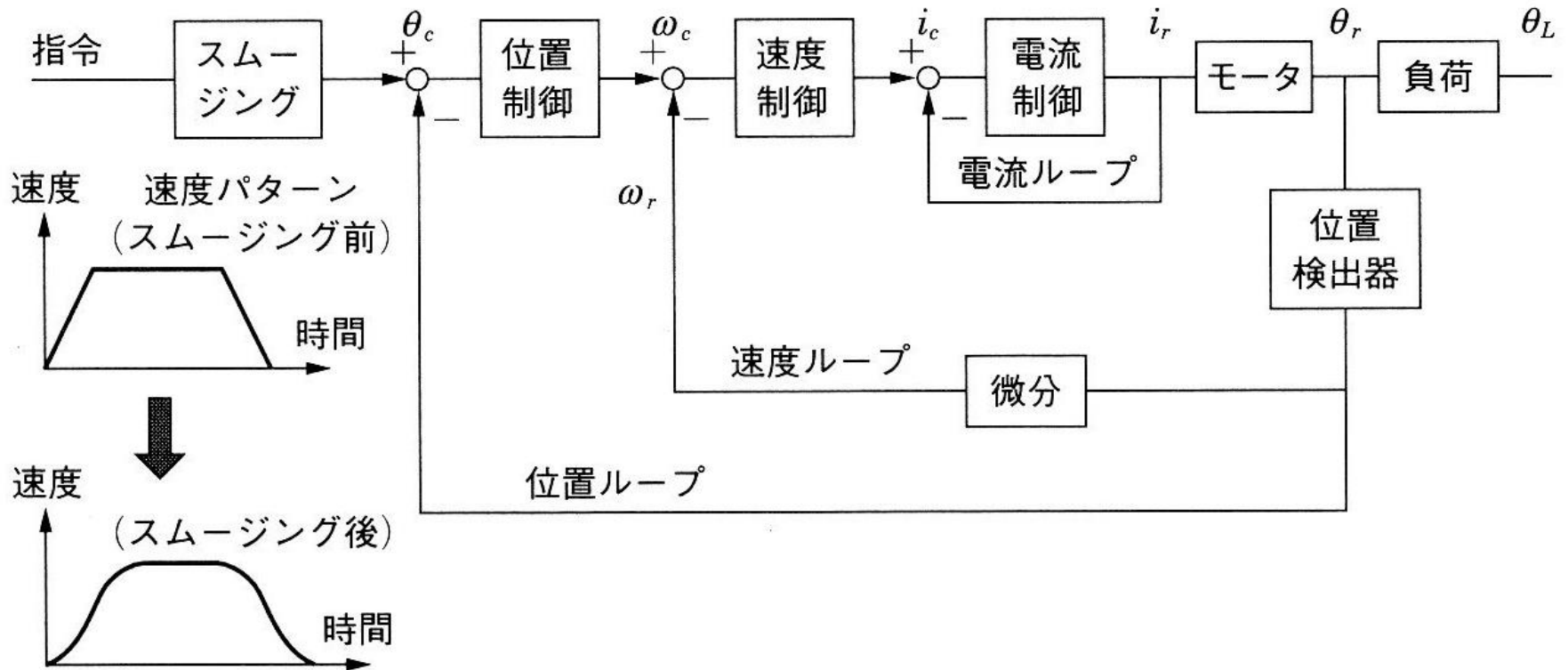


操作量 $u = K_p(\theta_c - \theta_e) + K_i \Sigma(\theta_c - \theta_e)\Delta t + K_d \frac{\Delta(\theta_c - \theta_e)}{\Delta t}$

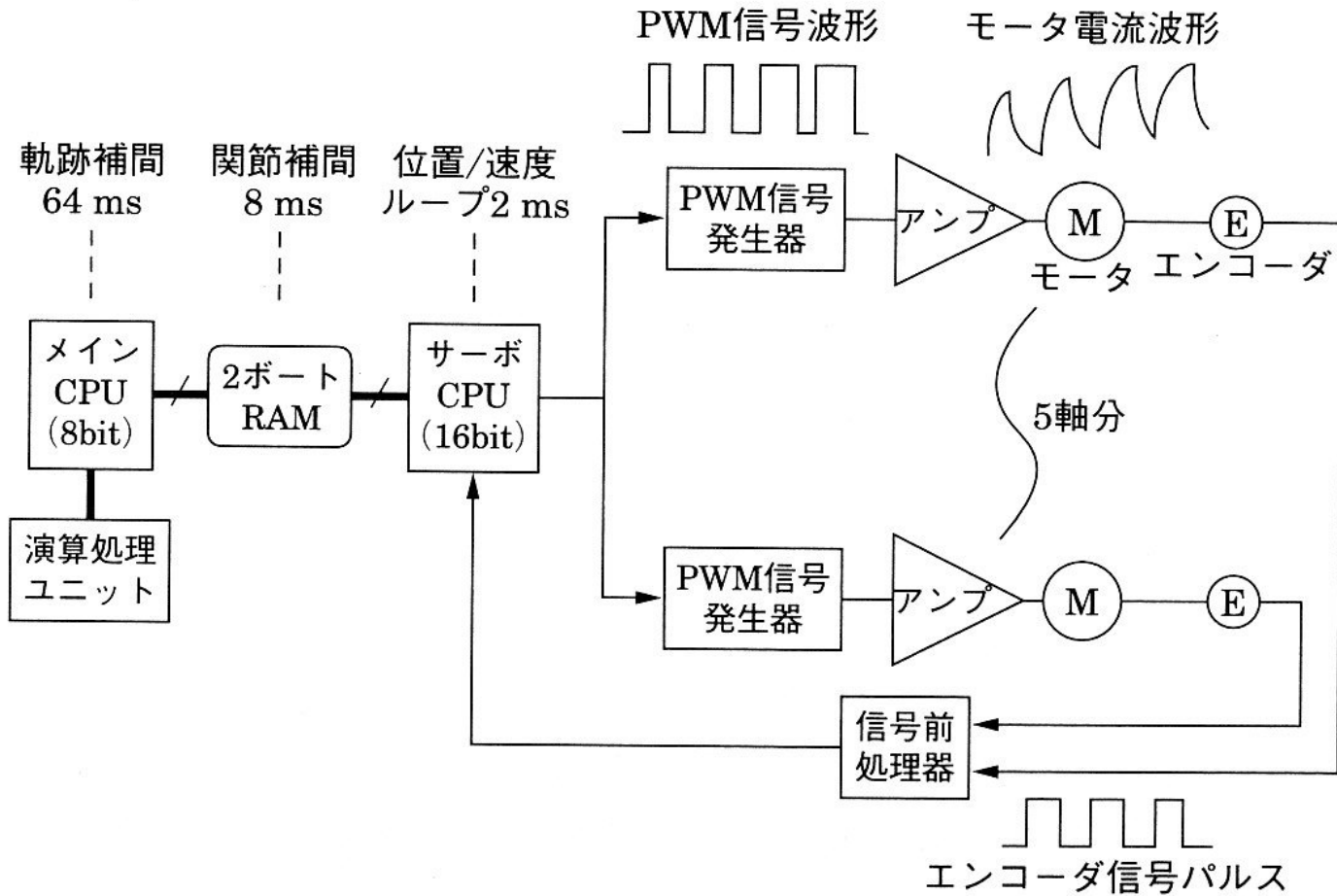
モータ電流指令、もしくはPWMデューティ比への反映

サーボ機構のブロック図 概略

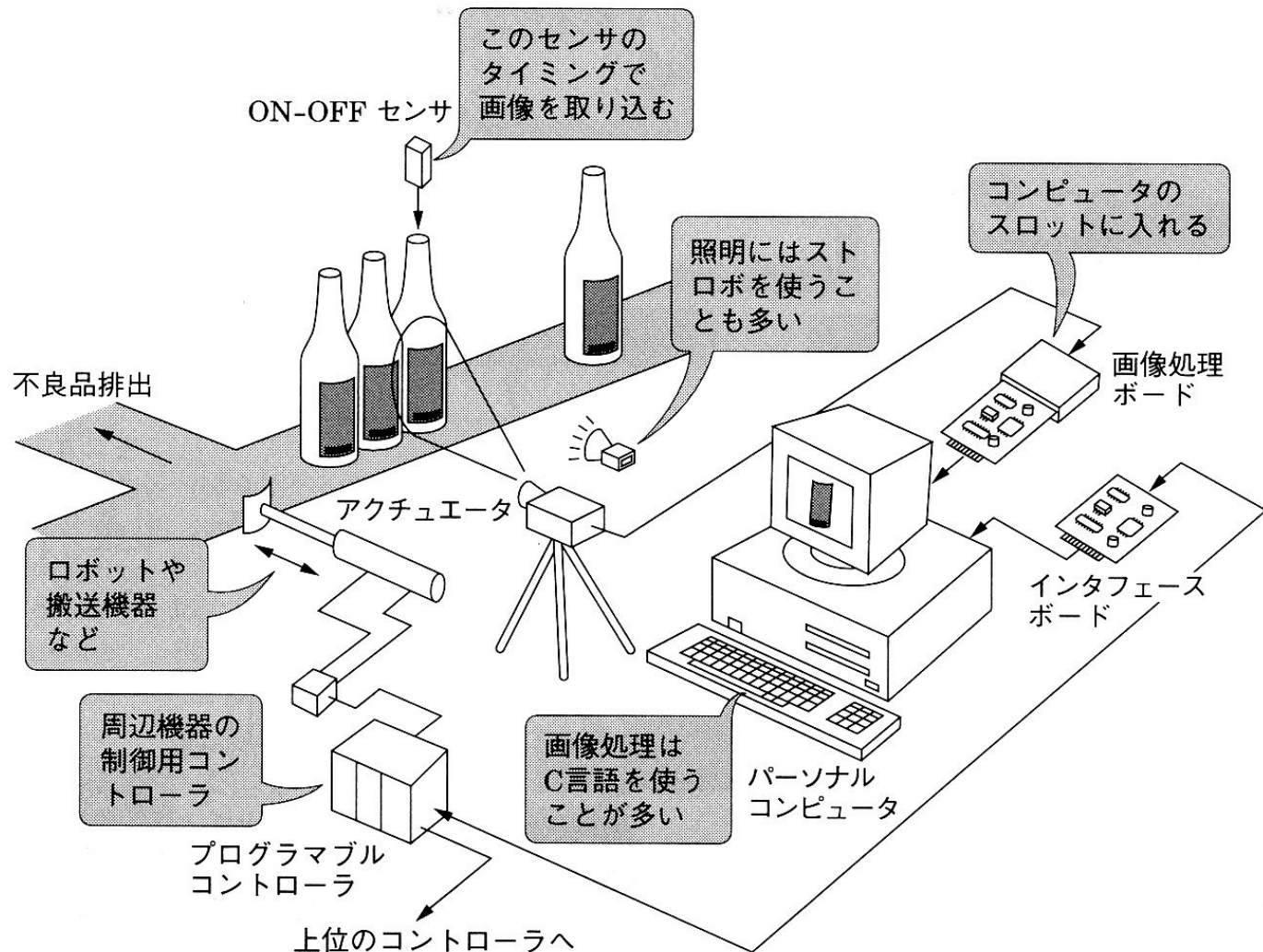
ω_c : 速度指令 i_c : 電流指令 θ_c : 位置指令 θ_L : 負荷位置
 ω_r : 速度帰還 i_r : 電流帰還 θ_r : 位置帰還



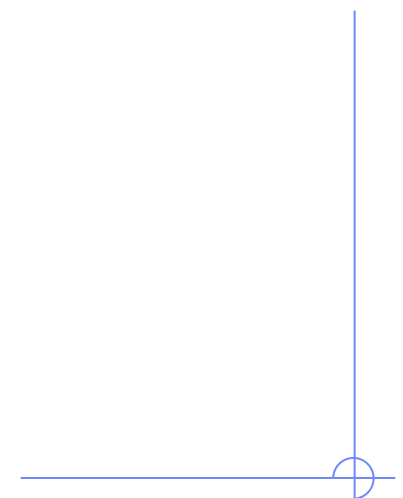
5軸マニピュレータの構成例



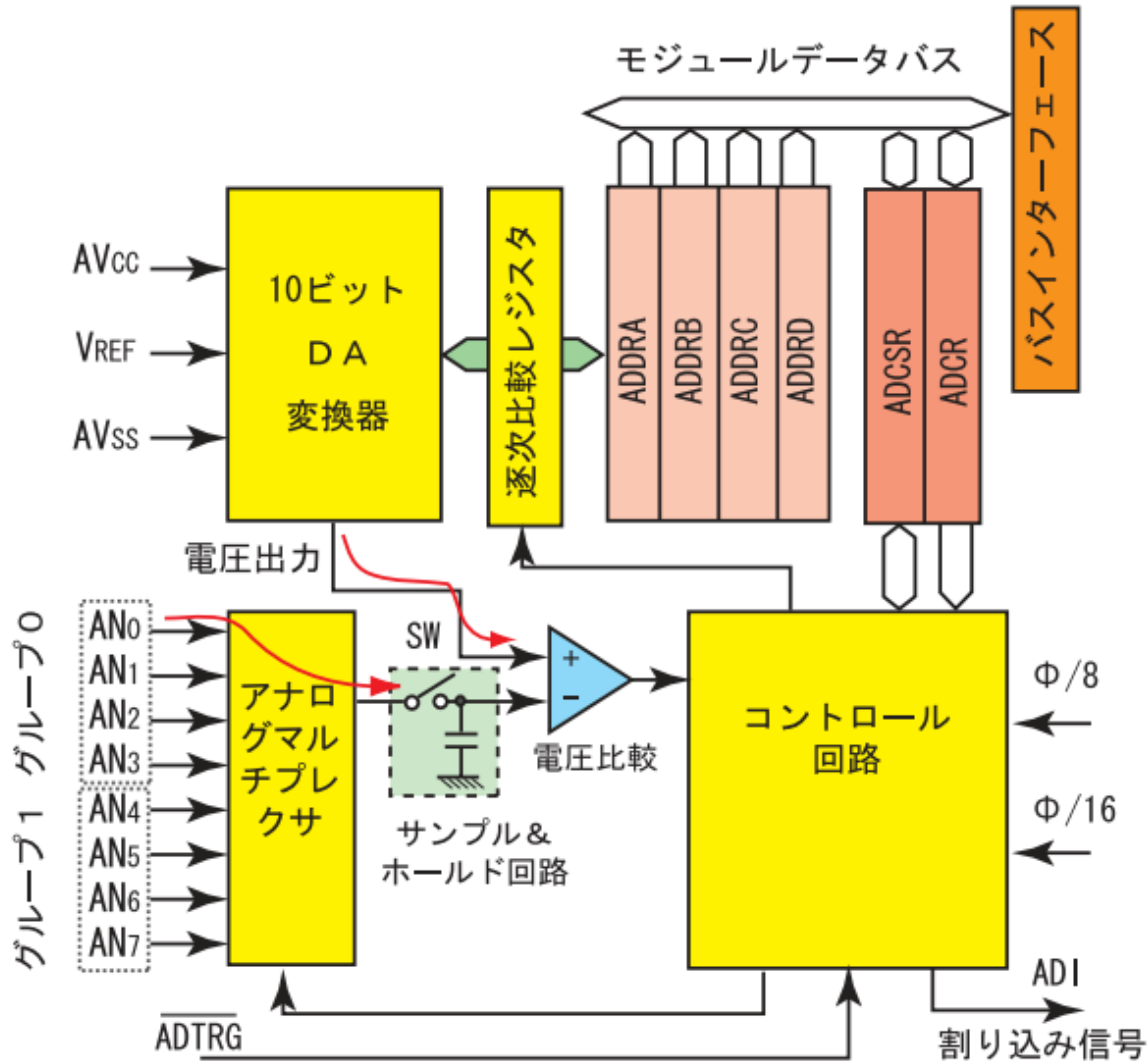
視覚センサを用いた検査装置



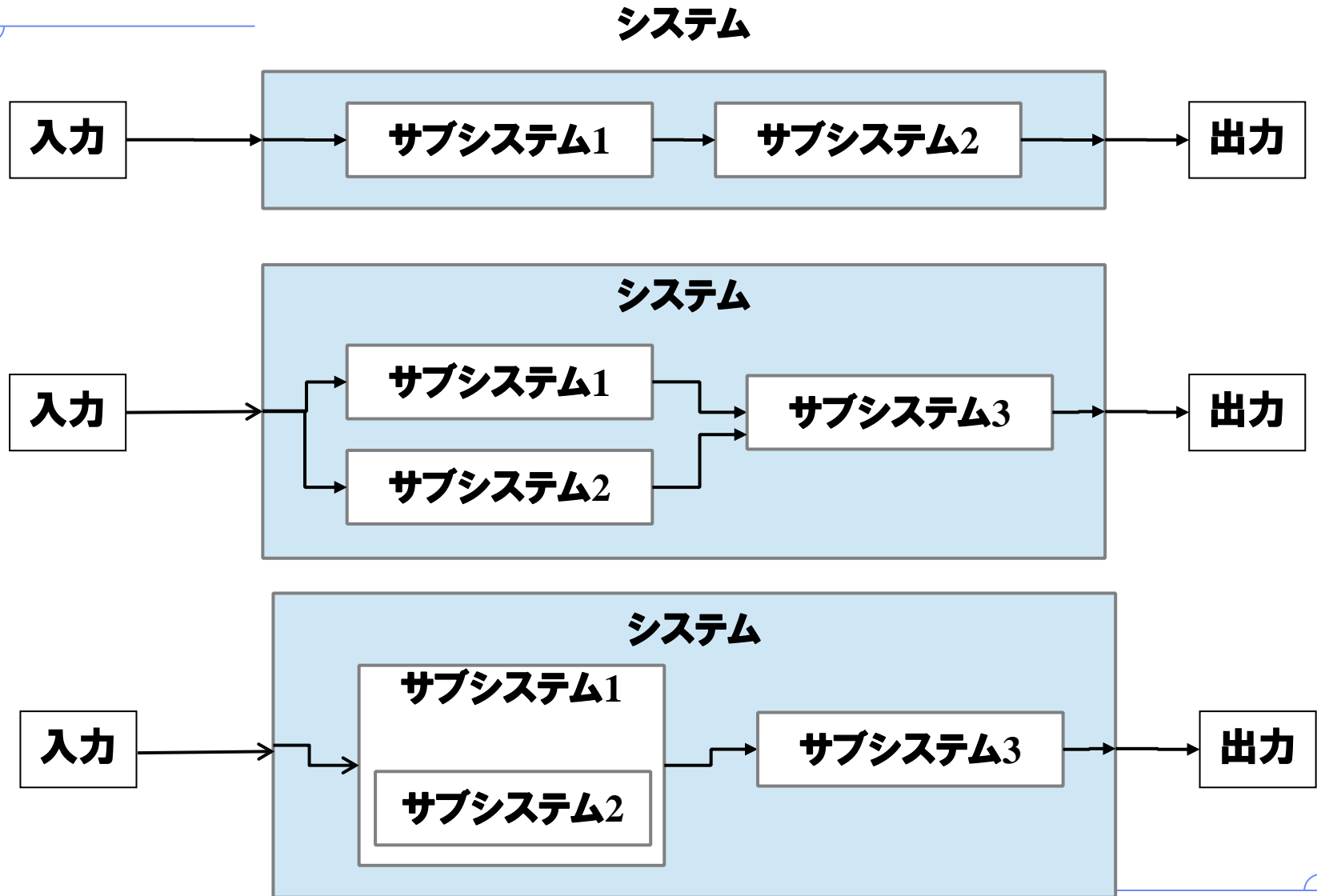
おわり



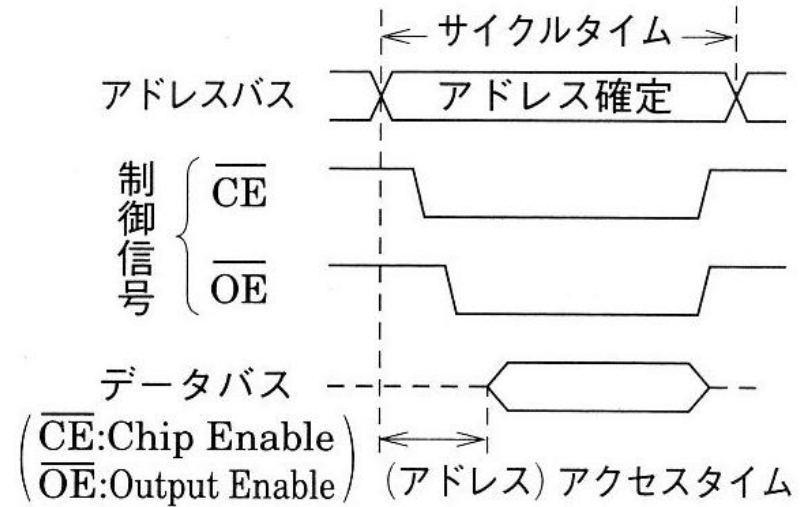
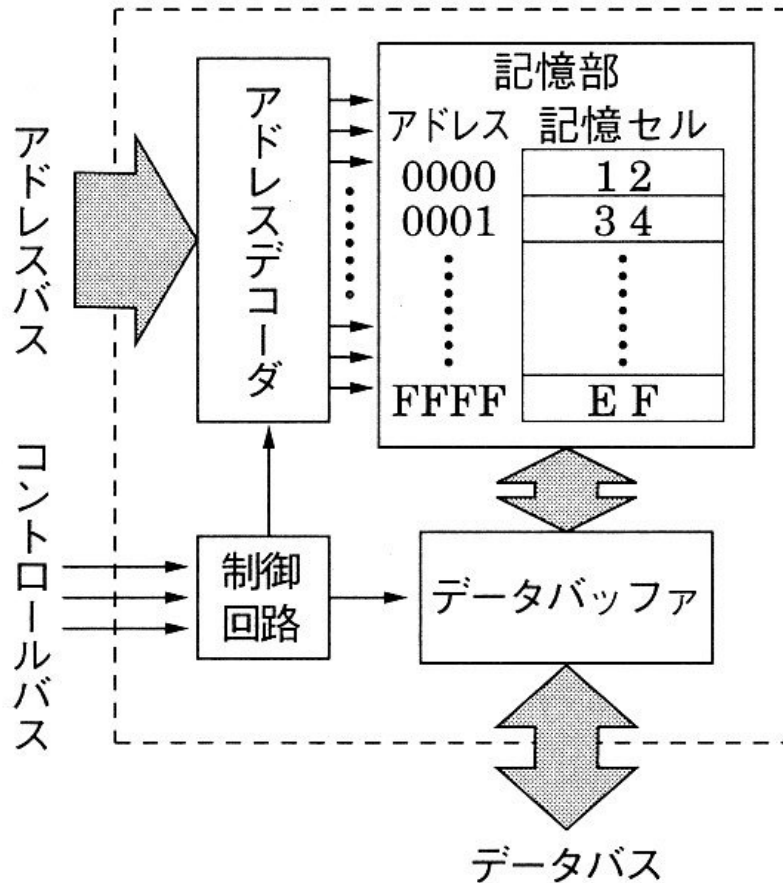
AD変換の例



システムの階層化



メモリの読み書き



**制御信号が入ったときのアドレスに応じたメモリに対して
データバス上でやりとりできる**